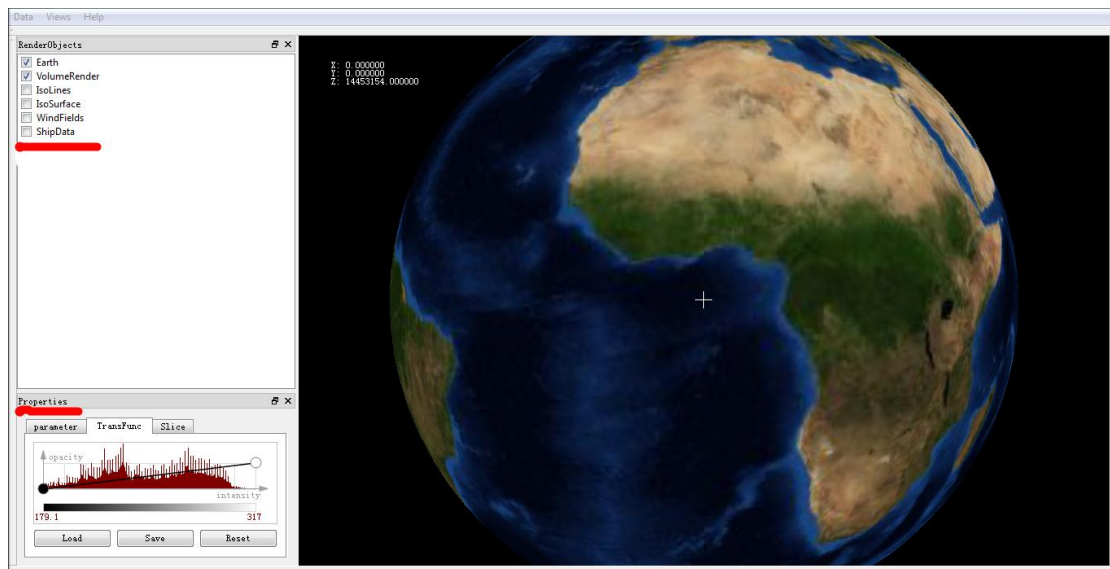


周报工作总结

肇昕 2012-12-17

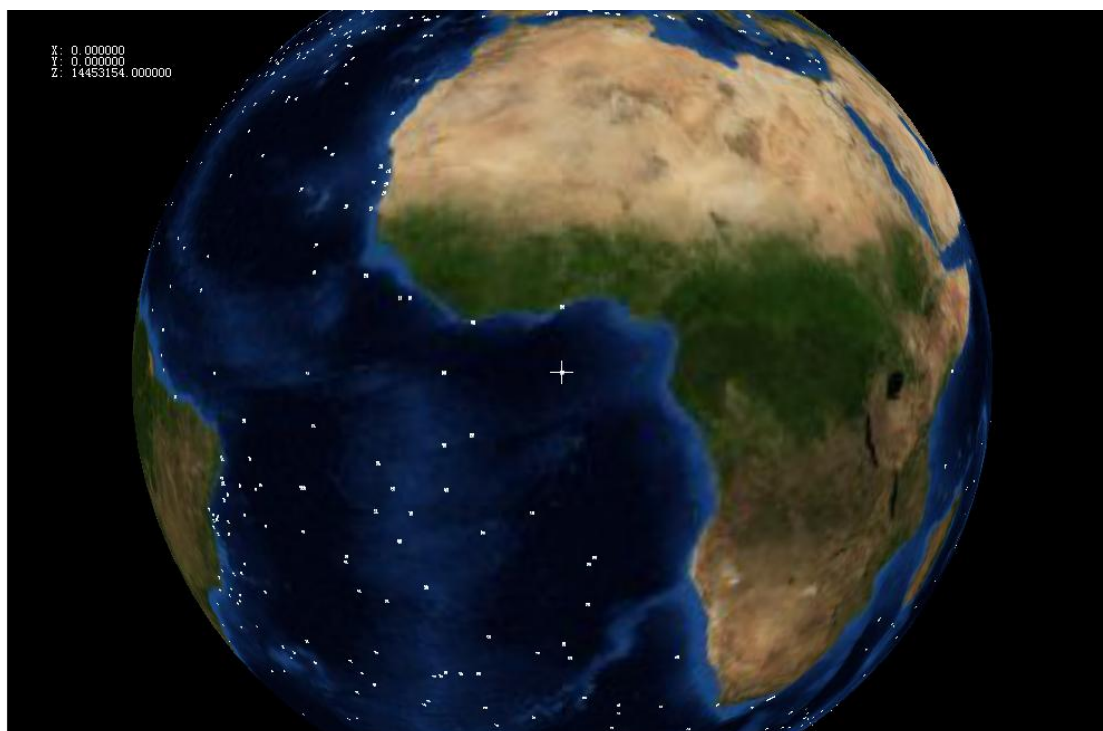
上周,我的主要工作是继续深化实现对于气象可视化中——常规数据可视化的框架实现,在上周中,我针对常规数据的基本数据结构、数据内容和数据的类型进行了详细的解析,具体得到的数据集中的具体变量的值进行了大概的分析。我将船舶数据、陆地数据和高空数据作为直接进行交互的三大类别。设计成为在地球上的基础框架上与风场、等值线等相平行关系的接口。

在后续的前端设计中,将会以属性(property)的形式添加各个类型的常规数据进去,目前只是作为一类数据的精细写法,其他的类型待一类成熟后可以模仿表现前端的设计。



(红色部分为基本选择界面和具体属性选择)

目前的实现效果为:



（图中的 icon 为读取的实际的传播数据的观测点的经度和纬度位置）

从图像中可以看到，传播信息分布在海洋和湖泊的位置，说明读取和转化成为三维坐标的算法的正确性，单独的一个数据源中提取了 5775 个数据点，在地球上表示详细的表示出来。

问题：数据点的分布在后期的交互中会加入时序——空间两种筛选条件，这样可以有效的减少同时显示在地球上的数据点。另外，希望可以开始一个 level 的关系，与 eye 相对于地球的位置进行匹配，可以做到远观 icon 较少，拉近增加显示的功能。

在编程过程中，目前采用的是 OpenGL 的基本绘制方法：

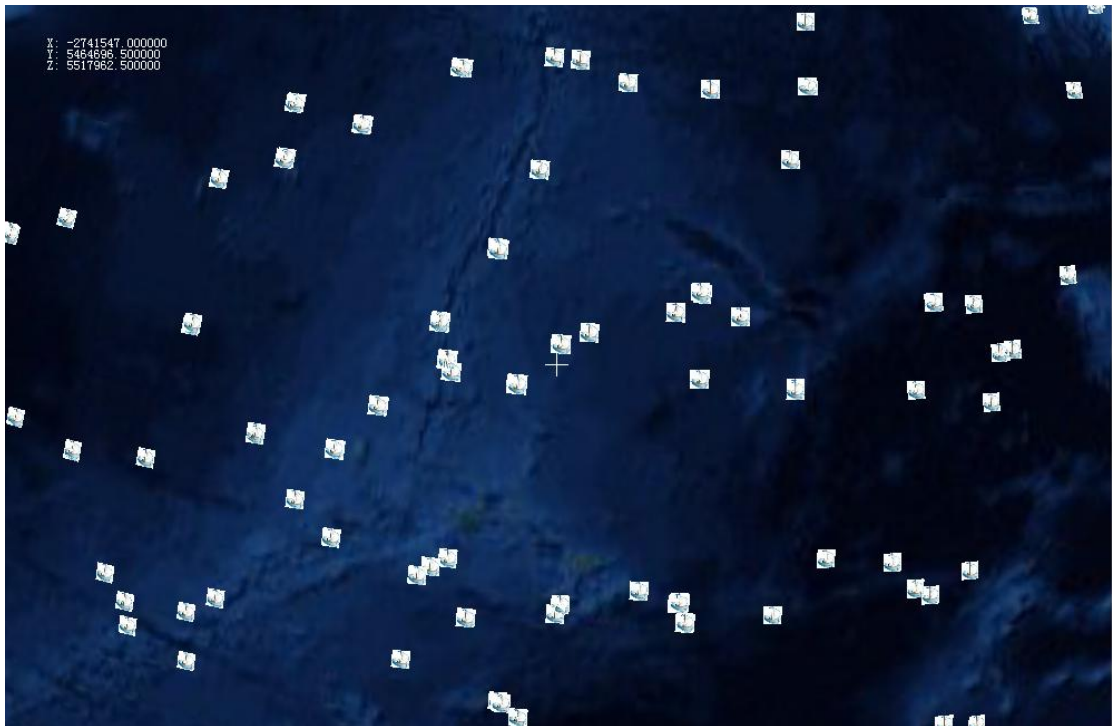
```
glBegin(GL_QUADS);  
glTexCoord2f(0,0);glVertex3f(-20000.0f, -20000.0f, 0.0f);  
glTexCoord2f(1,0);glVertex3f(20000.0f, -20000.0f, 0.0f);  
glTexCoord2f(1,1);glVertex3f(20000.0f, 20000.0f, 0.0f);  
glTexCoord2f(0,1);glVertex3f(-20000.0f, 20000.0f, 0.0f);  
glEnd();
```

目前速度上没有存在太大问题，读取的数据占用内存也不是很多，每个数据的大小为 2M 左右。但后续可能要更改为 shader 的绘制方式。

在每个 icon 的图标的显示方面，选用的是 billboard 的绘制形式，在实现过程之前，特地学习和查阅了 billboard 的原理和类别。

网址：http://nehe.gamedev.net/article/billboarding_how_to/18011/
<http://www.lighthouse3d.com/opengl/billboarding/index.php?billCyl>

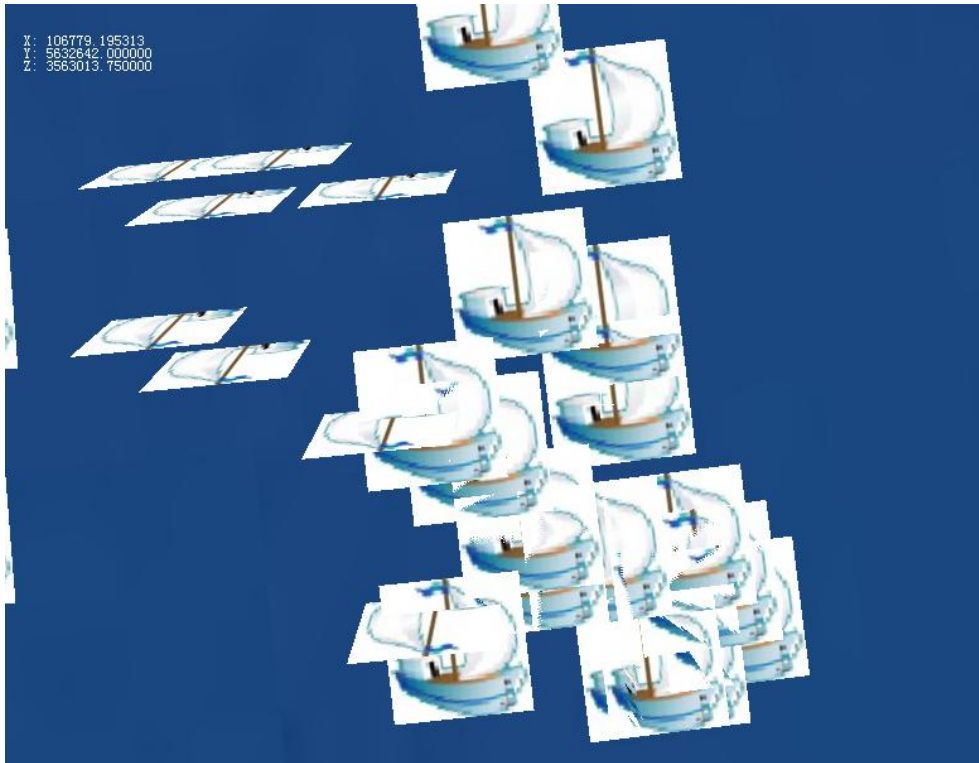
通过实际的测试，我选用的以球心点(0,0,0)为始终面向的点，icons 都面向球心的算法进行 billboard 绘制。如图：



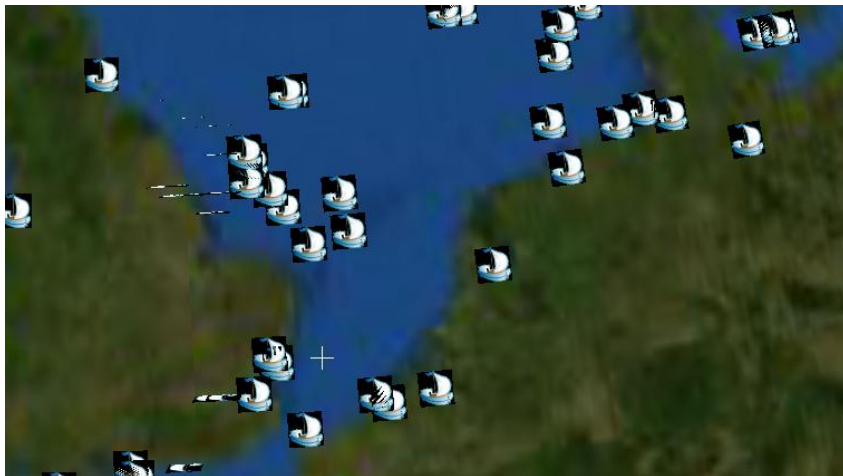
（每个 icon 都固定在地表，与地轴相垂直）

```
void BillboardSphericalBegin(float *cam, float *worldPos) {
    float lookAt[3]={0,0,1},objToCamProj[3],objToCam[3],upAux[3],angleCosine;
    objToCamProj[0] = cam[0] - worldPos[0] ;
    objToCamProj[1] = 0;
    objToCamProj[2] = cam[2] - worldPos[2] ;
    mathsNormalize(objToCamProj);
    mathsCrossProduct(upAux,lookAt,objToCamProj);
    angleCosine = mathsInnerProduct(lookAt,objToCamProj);
    if ((angleCosine < 0.99990) && (angleCosine > -0.9999))
        glRotatef(acos(angleCosine)*180/3.14,upAux[0], upAux[1], upAux[2]);
    objToCam[0] = cam[0] - worldPos[0] ;
    objToCam[1] = cam[1] - worldPos[1] ;
    objToCam[2] = cam[2] - worldPos[2] ;
    mathsNormalize(objToCam);
    angleCosine = mathsInnerProduct(objToCamProj,objToCam);
    if ((angleCosine < 0.99990) && (angleCosine > -0.9999))
        if (objToCam[1] < 0)
            glRotatef(acos(angleCosine)*180/3.14,1,0,0);
        else
            glRotatef(acos(angleCosine)*180/3.14,-1,0,0);
}
```

问题：



1. 在一些特殊点上会有变形。
2. icon 的贴图的透明问题



本周将会基于这两个问题进行解决，并且初步实现鼠标拾取功能。